Kybernetika

Tyrone Rees; Martin Stoll; Andy Wathen

All-at-once preconditioning in PDE-constrained optimization

Kybernetika, Vol. 46 (2010), No. 2, 341--360

Persistent URL: http://dml.cz/dmlcz/140748

KYBERNETIKA — VOLUME *46* (2010), NUMBER 2, PAGES 341–360

ALL-AT-ONCE PRECONDITIONING IN PDE-CONSTRAINED OPTIMIZATION

Tyrone Rees, Martin Stoll and Andy Wathen

The optimization of functions subject to partial differential equations (PDE) plays an important role in many areas of science and industry. In this paper we introduce the basic concepts of PDE-constrained optimization and show how the all-at-once approach will lead to linear systems in saddle point form. We will discuss implementation details and different boundary conditions. We then show how these system can be solved efficiently and discuss methods and preconditioners also in the case when bound constraints for the control are introduced. Numerical results will illustrate the competitiveness of our techniques.

Keywords: optimal control, preconditioning, partial differential equations

Classification: 65F10, 65N22, 65F50, 76D07

1. INTRODUCTION

For many decades researchers have studied the numerical solution of so-called forward PDE problems, where the solution to a PDE has to be computed. Using advances made for forward problems as well as the increase in available computing power has enabled researchers to look at inverse or design problems. In such problems the aim is to minimize a functional $J(y, u)$ subject to a PDE where y and u are the state and control of the optimality system, respectively.

Optimal control subject to PDEs is a field where many contributions were made over the last decade (see [17, 18, 28] for general introductions). We recently focussed our interest on the fast solution of the linear systems that arise when the discretized problem has to be solved [23, 24, 27]. In this paper we will focus our attention on the details of implementation, boundary conditions, etc. that we find often not addressed well.

2. THE PROBLEM

The optimization problem that we consider in this paper is given by the following setup: given function, $\bar{y}$, that represents the desired state, minimize the functional

$$J(y, u) := \frac{1}{2}\|y - \bar{y}\|^2_{L^2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L^2(\Omega)}, \tag{1}$$

for some domain $\Omega \in \mathbb{R}^d$ ($d = 2, 3$) with boundary Γ, where $\beta \in \mathbb{R}^+$ is a regularization parameter. The state y and the control u are linked via the state equation

$$\begin{cases} -\nabla^2 y = u & \text{in } \Omega, \\ \quad\;\; y = g & \text{on } \Gamma, \end{cases} \tag{2}$$

where g is a given function defined on the boundary Γ. Note that in our example the state equation is simply the Poisson equation with Dirichlet boundary data, but in general this could also be a more complicated PDE. Note that the variable y can be eliminated from (1) using the state equation (2) which would result in a problem where a function $J(y(u), u) := F(u)$ has to be minimized. Minimizing $F(u)$ is typically referred to as the reduced problem; for reasons that become clear in the next part we will refer to this problem as the unconstrained problem, since the function $F(u)$ has no additional constraints.

The case when $F(u)$ has to be minimized subject to so-called *bound constraints*

$$u_a(x) \leq u(x) \leq u_b(x) \text{ a.e in } \Omega \tag{3}$$

will be denoted as the constraint problem. Here, we define

$$\mathcal{U}_{ad} := \left\{ u \in L^2(\Omega) : u_a(x) \leq u(x) \leq u_b(x) \text{ a.e in } \Omega \right\}.$$

The presented setup can be summarized in the following PDE constrained optimization problem

$$\begin{cases} \min_{y,u} \frac{1}{2} \|y - \bar{y}\|^2_{L^2(\Omega)} + \frac{\beta}{2} \|u\|^2_{L^2(\Omega)} & \text{s.t.} \\ \qquad\qquad -\nabla^2 y = u & \text{in } \Omega \\ \qquad\qquad\qquad y = g & \text{on } \Gamma \\ \qquad u_a(x) \leq u(x) \leq u_b(x) & \text{a.e in } \Omega. \end{cases} \tag{4}$$

For a general discussion of problems of this type we refer to [9, 17, 23, 28] for more details.

There are two paths that we can now take to obtain the solution to the optimization problem (4). The first is *optimize-then-discretize* and the second approach is *discretize-then-optimize*. These two approaches coincide for the minimization of $J(y, u)$ subject to many PDE problems, including in particular the Poisson equation, but in general these two paths will not lead to the same setup (see [17] for a general discussion and [6] for a particular example where the state equation is given by the advection diffusion equation).

Following the discretize-then-optimize approach, we use the finite element method to discretize (4). Let $\{\phi_1, \ldots, \phi_{n+n_\partial}\}$ be a set of finite element basis functions associated with the n interior nodes and n_∂ boundary nodes of a triangulation of Ω. Consider $u_h = \sum_{i=1}^{n+n_\partial} \mathbf{u}_i \phi_i$ and $y_h = \sum_{i=1}^{n+n_\partial} \mathbf{y}_i \phi_i$, the discrete analogues of u and y respectively. Then it can be shown [23] that the discrete version of (4) is given by

$$\begin{cases} \min \frac{1}{2} \mathbf{y}^T M \mathbf{y} - \mathbf{b}^T \mathbf{y} + \frac{\beta}{2} \mathbf{u}^T M \mathbf{u} & \text{s.t.} \\ K\mathbf{y} = M\mathbf{u} - \mathbf{d} \\ \underline{\mathbf{u}}_{\mathbf{a}} \leq \mathbf{u} \leq \overline{\mathbf{u}}_{\mathbf{b}}, \end{cases} \tag{5}$$

where $\mathbf{d}$ represents the boundary data, $\mathbf{b}_i = \int \bar{y}\phi_i$ and $K_{i,j} = \int \nabla\phi_i \cdot \nabla\phi_j$, $M_{i,j} = \int \phi_i\phi_j$ represent the stiffness and mass matrices respectively. The third line in (5) are bounds on finite element expansion coefficients which follow easily from (3) for any Lagrange finite elements, at least if $\underline{\mathbf{u}}_\mathbf{a}$, $\overline{\mathbf{u}}_\mathbf{b}$ are piecewise constant on the chosen mesh. Here, the set of admissible controls is given by $U_{ad} := \{\mathbf{u} \in \mathbb{R}^n : \underline{\mathbf{u}}_\mathbf{a} \leq \mathbf{u} \leq \overline{\mathbf{u}}_\mathbf{b}\}$. Consider firstly the solution to the unconstrained problem, where no bound constraints on the control $\mathbf{u}$ are present. We can write down the Lagrangian

$$\mathcal{L}(\mathbf{y},\mathbf{u},\boldsymbol{\lambda}) = \frac{1}{2}\mathbf{y}^T M\mathbf{y} - \mathbf{b}^T\mathbf{y} + \frac{\beta}{2}\mathbf{u}^T M\mathbf{u} + \boldsymbol{\lambda}^T(M\mathbf{u} - K\mathbf{y} - \mathbf{d}), \tag{6}$$

where $\boldsymbol{\lambda}$ is a vector of Lagrange multipliers. From this we immediately obtain the following discrete optimality condition

$$\underbrace{\begin{bmatrix} M & 0 & -K \\ 0 & \beta M & M \\ -K & M & 0 \end{bmatrix}}_{\mathcal{K}} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ 0 \\ \mathbf{d} \end{bmatrix}. \tag{7}$$

In Section 4.1 we will discuss the efficient solution of the linear system (7).

The introduction of bound constraints adds an extra layer of complexity to the minimization of $J(\mathbf{y},\mathbf{u})$ as the control has to be kept within U_{ad} which requires an inner-outer iteration process. The optimization problem (5) can also be solved using a Lagrange multiplier approach (see [27, 28]). In this case the Lagrangian is also given by $\mathcal{L}(\mathbf{y},\mathbf{u},\boldsymbol{\lambda})$ as shown in (6). Here differention with respect to $\mathbf{y}$ and $\boldsymbol{\lambda}$ are as before, but when we consider box constraints on the control the last optimality condition becomes the complementarity conditions

$$(\mathbf{u} - \mathbf{u}^*)^T \nabla_\mathbf{u} L(\mathbf{y}^*,\mathbf{u}^*,\boldsymbol{\lambda}^*) = (\mathbf{u} - \mathbf{u}^*)^T(\beta M\mathbf{u}^* + M\boldsymbol{\lambda}^*) \geq 0 \;\; \forall\, \mathbf{u} \in U_{ad}, \tag{8}$$

where * denotes the optimal value of the variable. Condition (8) follows from the variational inequality

$$F'(\mathbf{u}^*)(\mathbf{u} - \mathbf{u}^*) \geq 0 \quad \forall\, \mathbf{u} \in U_{ad},$$

where $F(\mathbf{u})$ is equivalent to $J(\mathbf{y}(\mathbf{u}),\mathbf{u})$ and $U_{ad} = \{\mathbf{u} \in \mathbb{R}^n : \underline{\mathbf{u}}_\mathbf{a} \leq \mathbf{u} \leq \overline{\mathbf{u}}_\mathbf{b}\}$. Moreover, it follows that $\mathbf{u}^*$ solves the minimization problem

$$\min_{\mathbf{u} \in U_{ad}} \mathbf{u}^T(\beta M\mathbf{u}^* + M\boldsymbol{\lambda}^*) = (\mathbf{u}^*)^T(\beta M\mathbf{u}^* + M\boldsymbol{\lambda}^*). \tag{9}$$

With U_{ad} as defined above, we get the componentwise expression of $\mathbf{u}^*$

$$(\mathbf{u}^*)_i = \begin{cases} (\overline{\mathbf{u}}_\mathbf{b})_i & \text{if } (\beta M\mathbf{u}^* + M\boldsymbol{\lambda}^*)_i < 0 \\ \in U_{ad} & \text{if } (\beta M\mathbf{u}^* + M\boldsymbol{\lambda}^*)_i = 0 \\ (\underline{\mathbf{u}}_\mathbf{a})_i & \text{if } (\beta M\mathbf{u}^* + M\boldsymbol{\lambda}^*)_i > 0. \end{cases} \tag{10}$$

Relation (10) can be used to define a new Lagrangian by introducing two additional parameters $\boldsymbol{\mu}_a$ and $\boldsymbol{\mu}_b$ which enable us to define a new Lagrange function taking the bound constraints into account

$$\begin{aligned}\mathcal{L}(\mathbf{y},\mathbf{u},\boldsymbol{\lambda},\boldsymbol{\mu}_a,\boldsymbol{\mu}_b) := \ & \tfrac{1}{2}\mathbf{y}^T M\mathbf{y} - \mathbf{b}^T\mathbf{y} + \tfrac{\beta}{2}\mathbf{u}^T M\mathbf{u} + \boldsymbol{\lambda}^T(-K\mathbf{y} + M\mathbf{u} - \mathbf{d}) \\ & + \boldsymbol{\mu}_a^T\left(\underline{\mathbf{u}}_{\mathbf{a}} - \mathbf{u}\right) + \boldsymbol{\mu}_b^T\left(\mathbf{u} - \overline{\mathbf{u}}_{\mathbf{b}}\right),\end{aligned} \tag{11}$$

where $\boldsymbol{\mu}_a$ and $\boldsymbol{\mu}_b$ represent the Lagrange multipliers for the inequality constraints on $\mathbf{u}$. For the Lagrange function $\mathcal{L}(\mathbf{y},\mathbf{u},\boldsymbol{\lambda},\boldsymbol{\mu}_a,\boldsymbol{\mu}_b)$ the following theorem determines the optimality criteria of the system (5) (see [28, Theorem 1.4]).

Theorem 2.1. For an optimal control $\mathbf{u}^*$ with the corresponding state $\mathbf{y}^*$ and an invertible K there exist Lagrange multipliers $\boldsymbol{\lambda}$, $\boldsymbol{\mu}_a$, and $\boldsymbol{\mu}_b$ such that

$$\begin{aligned}&\nabla_{\mathbf{y}}\mathcal{L}(\mathbf{y}^*,\mathbf{u}^*,\boldsymbol{\lambda},\boldsymbol{\mu}_a,\boldsymbol{\mu}_b) = 0\\ &\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{y}^*,\mathbf{u}^*,\boldsymbol{\lambda},\boldsymbol{\mu}_a,\boldsymbol{\mu}_b) = 0\\ &\boldsymbol{\mu}_a \geq 0, \boldsymbol{\mu}_b \geq 0\\ &\boldsymbol{\mu}_a^T\left(\underline{\mathbf{u}}_{\mathbf{a}} - \mathbf{u}^*\right) = \boldsymbol{\mu}_b^T\left(\mathbf{u}^* - \overline{\mathbf{u}}_{\mathbf{b}}\right) = 0.\end{aligned}$$

The conditions given in Theorem 2.1 are the so-called *Karush–Kuhn–Tucker conditions* or *KKT conditions* (see [11, 20] for more information).

3. BOUNDARY CONDITIONS AND IMPLEMENTATION ISSUES

Homogeneous Dirichlet boundary

In practice, discretization of the optimality system (4) can be done by standard finite element methods if the boundary conditions that we discuss in this section are correctly implemented. In this section we want to address some of the issues that arise when using standard finite element packages such as dealii [1] for the discretization of (5).

In the normal manner with finite elements, we assemble the discretized Poisson's equation to give

$$\widehat{K}\mathbf{y} = \widehat{M}\mathbf{u} \tag{12}$$

before applying the essential boundary conditions. Implementation of the essential boundary conditions leads to

$$K\mathbf{y} = M\mathbf{u} - \mathbf{d}, \tag{13}$$

where $\mathbf{d}$ contains the Dirichlet boundary conditions.

To see how the boundary conditions are applied to the optimality system (7) we write out the stiffness matrix without having applied the essential boundary conditions as

$$\widehat{K} = \begin{bmatrix} X_K & Y_K^T \\ Y_K & K_I \end{bmatrix},$$

where the subscript I denotes the interior nodes. The corresponding mass matrix is

$$\widehat{M} = \begin{bmatrix} X_M & Y_M^T \\ Y_M & M_I \end{bmatrix}.$$

So far the Dirichlet boundary conditions have not been applied and we will discuss this now in more detail. The discrete optimality system in the unconstrained case and with no essential boundary condition is given by

$$\begin{bmatrix} X_M & Y_M^T & 0 & 0 & -X_K^T & -Y_K \\ Y_M & M_I & 0 & 0 & -Y_K^T & -K_I^T \\ 0 & 0 & \beta X_M & \beta Y_M^T & X_M & Y_M^T \\ 0 & 0 & \beta Y_M & \beta M_I & Y_M & N_I \\ -X_K & -Y_K^T & X_M & Y_M^T & 0 & 0 \\ -Y_K & -K_I & Y_M & M_I & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}_B \\ \mathbf{y}_I \\ \mathbf{u}_B \\ \mathbf{u}_I \\ \boldsymbol{\lambda}_B \\ \boldsymbol{\lambda}_I \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{b}_B \\ \mathbf{b}_I \end{bmatrix}. \tag{14}$$

Consider first the case where homogeneous Dirichlet boundary conditions are given for the state and hence for the adjoint variable, i. e., $\mathbf{y}_B = 0$ and $\boldsymbol{\lambda}_B = 0$ (see [28, Lemma 2.24]). Applying this boundary condition we get

$$\begin{bmatrix} 0 & 0 & 0 & 0 & -I & 0 \\ 0 & M_I & 0 & 0 & 0 & -K_I^T \\ 0 & 0 & \beta X_M & \beta Y_M^T & 0 & Y_M^T \\ 0 & 0 & \beta Y_M & \beta M_I & 0 & M_I \\ -I & 0 & 0 & 0 & 0 & 0 \\ 0 & -K_I & Y_M & M_I & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}_B \\ \mathbf{y}_I \\ \mathbf{u}_B \\ \mathbf{u}_I \\ \boldsymbol{\lambda}_B \\ \boldsymbol{\lambda}_I \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b}_I \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Note that we still need to know the value of $\mathbf{u}$ on the boundary. Consider the third and fourth equations of (14) written as

$$0 = M(\beta \mathbf{u} + \boldsymbol{\lambda}) = \begin{bmatrix} X_M & Y_M^T \\ Y_M & M_I \end{bmatrix} \begin{bmatrix} \beta \mathbf{u}_B + \boldsymbol{\lambda}_B \\ \beta \mathbf{u}_I + \boldsymbol{\lambda}_I \end{bmatrix},$$

or, equivalently,

$$0 = X_M(\beta \mathbf{u}_B + \boldsymbol{\lambda}_B) + Y_M^T(\beta \mathbf{u}_I + \boldsymbol{\lambda}_I) \tag{15}$$

$$0 = Y_M(\beta \mathbf{u}_B + \boldsymbol{\lambda}_B) + M_I(\beta \mathbf{u}_I + \boldsymbol{\lambda}_I). \tag{16}$$

Equation (16) gives $(\beta \mathbf{u}_I + \boldsymbol{\lambda}_I) = -M_I^{-1} Y_M (\beta \mathbf{u}_B + \boldsymbol{\lambda}_B)$ and putting that into (15) we get

$$0 = \left[X_M - Y_M^T M_I^{-1} Y_M\right](\beta \mathbf{u}_B + \boldsymbol{\lambda}_B).$$

Since $\widehat{M}$ is positive definite, its Schur-complement $X_M - Y_M^T M_I^{-1} Y_M$ is invertible and so $\beta \mathbf{u}_B + \boldsymbol{\lambda}_B = 0$. Since $\boldsymbol{\lambda}_B = 0$ in this case, we have also that $\mathbf{u}_B = 0$.

Finally, applying this boundary condition, we get the linear system

$$\begin{bmatrix} 0 & 0 & 0 & 0 & -I & 0 \\ 0 & M_I & 0 & 0 & 0 & -K_I^T \\ 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta M_I & 0 & M_I \\ -I & 0 & 0 & 0 & 0 & 0 \\ 0 & -K_I & 0 & M_I & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}_B \\ \mathbf{y}_I \\ \mathbf{u}_B \\ \mathbf{u}_I \\ \boldsymbol{\lambda}_B \\ \boldsymbol{\lambda}_I \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b}_I \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{17}$$

Going back to the fact that we want to employ standard finite element packages that are freely available, it has to be noticed that (17) would typically be taken to be

$$\begin{bmatrix} \mathbf{I} & 0 & 0 & 0 & -I & 0 \\ 0 & M_I & 0 & 0 & 0 & -K_I^T \\ 0 & 0 & \beta I & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \beta M_I & 0 & M_I \\ -I & 0 & \mathbf{I} & 0 & 0 & 0 \\ 0 & -K_I & 0 & M_I & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}_B \\ \mathbf{y}_I \\ \mathbf{u}_B \\ \mathbf{u}_I \\ \boldsymbol{\lambda}_B \\ \boldsymbol{\lambda}_I \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b}_I \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{18}$$

in such a software package. Note that (18) and (17) only yield the same solutions if the components of $\mathbf{y}$, $\boldsymbol{\lambda}$ and $\mathbf{u}$ on the boundary are zero. This has to be taken into account when one wants to solve problems of this type employing standard finite element packages. One possibility would be to only work with a saddle point problem on the inner nodes of the domain but this would mean for a practical application that the matrices need to be condensed; this is not feasible for most applications as storage requirements and computing times would be too expensive. One way to achieve the same effect is to use Krylov subspace solvers such as MINRES [21] or Conjugate Gradients (CG) [15] which compute an approximation, satisfying certain optimality criteria (cf. [8, 14, 25]), to the solution in the subspace

$$K_k(\mathcal{K}, \mathbf{r}_0) = \operatorname{span}\left\{\mathbf{r}_0, \mathcal{K}\mathbf{r}_0, \mathcal{K}^2\mathbf{r}_0, \ldots, \mathcal{K}^{k-1}\mathbf{r}_0\right\},$$

where $\mathbf{r}_0$ is the initial residual. These methods only require the multiplication of the saddle point matrix $\mathcal{K}$ with a vector. In order to use the matrices coming from a finite element package as given in (18) we have to start off with the initial residual $\mathbf{r}_0$ to have zero components in the direction of the boundary values for $\mathbf{y}$, $\boldsymbol{\lambda}$ and $\mathbf{u}$. This means that the bold identity matrices in equation (18) will be mapped onto zeros whenever a matrix-vector multiplication has to be performed. If this is satisfied it is easy to see that the solutions to (18) and (17) will be equivalent. For the case of homogeneous Dirichlet boundary conditions we know that both state $\mathbf{y}$ and adjoint variable $\boldsymbol{\lambda}$ are zero on the boundary. In addition, we have already shown that the control $\mathbf{u}$ is zero on the boundary and any Krylov subspace solver will provide an approximation to the solution of (17) as all boundary components will be zero.

Inhomogeneous Dirichlet boundary

For the case of inhomogeneous boundary conditions on the state $\mathbf{y}$, it is not immediately obvious what the right boundary conditions are for the adjoint variable $\boldsymbol{\lambda}$ and hence for the control $\mathbf{u}$. We will discuss this now in more detail by considering the inhomogeneous problem

$$\begin{cases} \min_{y,u} J(y,u) & \text{s.t.} \\ \quad -\nabla^2 y = u & \text{in } \Omega \\ \quad y = g & \text{on } \Gamma. \end{cases} \tag{19}$$

Recall that in this example the discretize-then-optimize and optimize-then-discretize approaches are equivalent, so the boundary conditions of the adjoint equation in the continuous setting will be the same as those needed for the corresponding equation in the discretize-then-optimize approach we adopt above.

We introduce two Lagrange multipliers, λ_1 and λ_2, and formally consider the Lagrangian

$$\mathcal{L} := \frac{1}{2}\|y-\bar{y}\|^2_{L^2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L^2(\Omega)} - \int_\Omega (-\nabla^2 y - u)\lambda_1 \,\mathrm{d}x - \int_\Gamma (y-g)\lambda_2 \,\mathrm{d}s.$$

Now consider the Fréchet derivative with respect to y in the direction h:

$$\begin{aligned} D_y\mathcal{L}(y,u,\lambda_1,\lambda_2)h &= \int_\Omega (y-\bar{y})h\,\mathrm{d}x - \int_\Omega -\nabla^2 h\,\lambda_1\,\mathrm{d}x - \int_\Gamma \lambda_2 h\,\mathrm{d}s \\ &= \int_\Omega (y-\bar{y})h\,\mathrm{d}x + \int_\Omega h\,\nabla^2\lambda_1\,\mathrm{d}x - \int_\Gamma \frac{\partial h}{\partial n}\lambda_1\,\mathrm{d}s \\ &\quad + \int_\Gamma h\frac{\partial \lambda_1}{\partial n}\,\mathrm{d}s - \int_\Gamma \lambda_2 h\,\mathrm{d}s. \end{aligned} \tag{20}$$

For a minimum, since there are no restrictions (e. g. box constraints) on the state, we must have that

$$D_y\mathcal{L}(y,u,\lambda_1,\lambda_2)h = 0 \quad \forall\, h \in H^1(\Omega).$$

In particular, we must have $D_y\mathcal{L}(y,u,\lambda_1,\lambda_2)h = 0$ for all $h \in C_0^\infty(\Omega)$. In this case $h|_\Gamma = 0 = \frac{\partial h}{\partial n}|_\Gamma$, and so (20) reduces to

$$\int_\Omega (y - \bar{y} + \nabla^2\lambda_1)h\,\mathrm{d}x = 0 \quad \forall\;\, h \in C_0^\infty(\Omega).$$

Thus, applying the fundamental lemma of the Calculus of Variations, we get that

$$-\nabla^2\lambda_1 = y - \bar{y} \quad \text{in } \Omega. \tag{21}$$

Now consider $h \in H_0^1(\Omega)$, so that $h|_\Gamma = 0$. Then we get

$$\int_\Gamma \frac{\partial h}{\partial n}\lambda_1\,\mathrm{d}s = 0 \quad \forall\, h \in H_0^1(\Omega)$$

so we have

$$\lambda_1 = 0 \quad \text{on } \Gamma. \tag{22}$$

The remaining equations give us the link between λ_1 and λ_2, namely

$$\lambda_1 = \frac{\partial \lambda_2}{\partial n} \quad \text{on } \Omega.$$

If we ignore the index for λ_1 we can write the adjoint equation as

$$\begin{aligned} -\nabla^2\lambda &= y - \bar{y} \quad \text{in } \Omega \qquad (23)\\ \lambda &= 0 \qquad\;\; \text{on } \Gamma. \qquad (24) \end{aligned}$$

For completeness we derive the continuous variational (in)equality. In the case without constraints on the control we get that $D_u\mathcal{L}(y,u,\lambda_1,\lambda_2)h = 0$, so this gives us

$$\int_\Omega (\beta u + \lambda)h \, \mathrm{d}x = 0 \quad \forall\, h \in H^1(\Omega).$$

Using the fundamental lemma of the calculus of variations, we get

$$\beta u + \lambda = 0 \quad \text{in } \Omega. \tag{25}$$

Based on (23), (24) and (25) the control u is defined on the boundary of Ω. Again, employing finite elements for the discretization of the optimal control problem

$$\begin{cases} \min \frac{1}{2}\left\|y-\bar{y}\right\|^2_{L^2(\Omega)} + \frac{\beta}{2}\left\|u\right\|^2_{L^2(\Omega)} & \text{s.t.} \\ -\nabla^2 y = u & \text{in } \Omega \\ y = g & \text{on } \Gamma, \end{cases} \tag{26}$$

we obtain the following

$$\begin{bmatrix} 0 & 0 & 0 & 0 & -I & 0 \\ 0 & M_I & 0 & 0 & 0 & -K_I^T \\ 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta M_I & 0 & M_I \\ -I & 0 & 0 & 0 & 0 & 0 \\ 0 & -K_I & 0 & M_I & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}_B \\ \mathbf{y}_I \\ \mathbf{u}_B \\ \mathbf{u}_I \\ \boldsymbol{\lambda}_B \\ \boldsymbol{\lambda}_I \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b}_I - Y_M \mathbf{g}_B \\ 0 \\ 0 \\ \mathbf{g}_B \\ Y_K \mathbf{g}_B \end{bmatrix} \tag{27}$$

where $\mathbf{g}_B$ interpolates $-g$ on Γ and both $\mathbf{u}_B$ and $\boldsymbol{\lambda}_B$ are zero on the boundary. For consistency with the notation used above we define $\widehat{\mathbf{b}}_I := \mathbf{b}_I - Y_M\mathbf{g}_B$ and $\mathbf{d}_I := Y_K\mathbf{g}_B$. Again, the same implementation issues arise as the finite element package of choice will most likely only present a system of the form

$$\begin{bmatrix} \mathbf{I} & 0 & 0 & 0 & -I & 0 \\ 0 & M_I & 0 & 0 & 0 & -K_I^T \\ 0 & 0 & \beta I & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & \beta M_I & 0 & M_I \\ -I & 0 & \mathbf{I} & 0 & 0 & 0 \\ 0 & -K_I & 0 & M_I & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}_B \\ \mathbf{y}_I \\ \mathbf{u}_B \\ \mathbf{u}_I \\ \boldsymbol{\lambda}_B \\ \boldsymbol{\lambda}_I \end{bmatrix} = \begin{bmatrix} 0 \\ \widehat{\mathbf{b}}_I \\ 0 \\ 0 \\ \mathbf{g}_B \\ \mathbf{d}_I \end{bmatrix}. \tag{28}$$

Note that this system could be decoupled to give two independent linear systems – one involving the boundary terms, $[\mathbf{y}_B, \mathbf{u}_B, \boldsymbol{\lambda}_B]^T$, and the other in the interior terms, $[\mathbf{y}_I, \mathbf{u}_I, \boldsymbol{\lambda}_I]^T$. Solving only with the second of these would give the solution of the optimal control problem, since the boundary values are known; however, as described above, deflating the matrix in this way would be an expensive operation. Consider the decoupled system containing just the boundary terms of (18):

$$\begin{bmatrix} \mathbf{I} & 0 & -I \\ 0 & \beta I & \mathbf{I} \\ -I & \mathbf{I} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}_B \\ \mathbf{u}_B \\ \boldsymbol{\lambda}_B \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{g}_B \end{bmatrix}. \tag{29}$$

When the bold identity matrices – which are an artifact of the finite element software – are included it is easy to see that this system will yield incorrect values for $\mathbf{y}_B, \mathbf{u}_B$, and $\boldsymbol{\lambda}_B$. If we set $\mathbf{g}_B = 0$ this will give the solution $\mathbf{y}_B = \mathbf{u}_B = \boldsymbol{\lambda}_B = 0$, as in the homogeneous case. Since the interior and boundary equations in (28) are independent, and the boundary conditions are still present in $\widehat{\mathbf{b}}_I$ and $\mathbf{d}_I$, this will not affect the solution at the interior nodes. After having obtained the solution we can then set the correct boundary values for the optimal state and control. We will show results using this technique in Section 5.

3.1. Other boundary conditions

Consider the following problem, with mixed boundary conditions:

$$\left\{ \begin{array}{c} \min \frac{1}{2} \|y - \bar{y}\|^2_{L^2(\Omega)} + \frac{\beta}{2} \|u\|^2_{L^2(\Omega)} \quad \text{s.t.} \\ -\nabla^2 y = u \quad \text{in } \Omega \\ y = g_1 \text{ on } \Gamma_1 \quad \text{and} \quad \frac{\partial y}{\partial n} = g_2 \text{ on } \Gamma_2 \\ u_a(x) \le u(x) \le u_b(x) \quad \text{a.e in } \Omega, \end{array} \right. \tag{30}$$

where $\Gamma_1 \cap \Gamma_2 = \emptyset$ and $\Gamma_1 \cup \Gamma_2 = \Gamma$. This is a generalization of the case considered above: $\Gamma_1 = \Gamma$ gives the inhomogeneous Dirichlet problem. Problems of this type can be treated in the same way as described above. In this case we simply split up the matrices K and M before applying the boundary conditions as

$$K = \begin{bmatrix} X_K & Y_K^T \\ Y_K & K_N \end{bmatrix}, \quad M = \begin{bmatrix} X_M & Y_M^T \\ Y_M & M_N \end{bmatrix},$$

where K_N and M_N refer to the nodes in the interior of Ω, as in the purely Dirichlet case, with the addition of the nodes on Γ_2. In the case where $g_2 \neq 0$ there will also be the standard addition of an integral over Γ_2 on the right hand side which is associated with the Neumann boundary condition.

In the special case where $\Gamma_1 = \emptyset$ we have a purely Neumann problem, and hence K is a singular matrix with a one dimensional kernel. In the numerical methods that follow we require K to be invertible, so we fix $\mathbf{y}$ at one node, in effect giving us a mixed problem with a Dirichlet boundary condition at just one node. The stiffness matrix K is then invertible, and we can use the same method as described above for the mixed boundary condition case.

4. NUMERICAL SOLUTION

4.1. Without bound constraints

The numerical solution of the optimality system (5) based on the Lagrange multiplier approach shown in Section 2 leads to solving the linear system given in (7). The system matrix $\mathcal{K}$ is symmetric and indefinite and in so-called saddle point form. Linear systems of this type have been studied intensively and we refer to [2, 8] for a general discussion of solution techniques. Here we will focus on the systems that arise in the context of PDE-constrained optimization. Systems of the type given in

(7) are typically very poorly conditioned which means any iterative solver will only be used in conjunction with preconditioning techniques, i. e.,

$$\mathcal{P}^{-1}\mathcal{K}\mathbf{x} = \mathcal{P}^{-1}\mathbf{b},$$

where the preconditioner $\mathcal{P}$ represents a good approximation to $\mathcal{K}$ but should also be easy to invert. The question now is which preconditioner is best suited for the linear system (7).

One 'ideal' preconditioner, proposed by Rees, Dollar and Wathen in [23], is the block-diagonal preconditioner

$$\widetilde{\mathcal{P}}_{BD} = \begin{bmatrix} M & 0 & 0 \\ 0 & \beta M & 0 \\ 0 & 0 & KM^{-1}K^T \end{bmatrix},$$

where, in the (3,3) block, the Schur complement $KM^{-1}K^T + \frac{1}{\beta}M$ – which would give the exact solution in three iterations using MINRES [19] – has been approximated by its dominant part, $KM^{-1}K^T$.

This preconditioner is symmetric and positive definite, which allows us to use the minimal residual method (MINRES) proposed in [21], a method designed for symmetric and indefinite systems. However, preconditioned MINRES requires a solve with $\widetilde{\mathcal{P}}_{BD}$ at each step of the iteration, which is expensive here. Therefore, we approximate $\widetilde{\mathcal{P}}_{BD}$ with

$$\mathcal{P}_{BD} = \begin{bmatrix} A_0 & 0 & 0 \\ 0 & \beta A_0 & 0 \\ 0 & 0 & S_0 \end{bmatrix},$$

where A_0 and S_0 are matrices – possibly defined implicitly – that are spectrally equivalent to M and $KM^{-1}K^T$ respectively and which are inexpensive to solve for a given right hand side.

Consider first the action of the inverse of the mass matrix; suppose we have a system $M\mathbf{z} = \hat{\mathbf{b}}$. One choice of preconditioner for the mass matrix is to let A_0^{-1} denote a fixed number of steps of the Chebyshev semi-iteration [12, 13] used to accelerate a relaxed Jacobi iteration. This is given by the three term recurrence relation

$$\mathbf{z}^{(k+1)} = \vartheta_{k+1}(S\mathbf{z}^{(k)} + \mathbf{g} - \mathbf{z}^{(k-1)}) + \mathbf{z}^{(k-1)}, \tag{31}$$

where $S = I - \omega D^{-1}M$, $\mathbf{g} = \omega D^{-1}\hat{\mathbf{b}}$, $\vartheta_{k+1} = \frac{T_k(1/\rho)}{\rho T_{k+1}(1/\rho)}$ and T_k denotes the k^{th} Chebyshev polynomial of the first kind (see Algorithm (4.1)). The relaxation parameter ω is chosen in such a way that the eigenvalues of S lie in an interval which is symmetric about the origin (see [8]). As shown by Wathen and Rees [30] this method is very effective in the case of the mass matrix since the values ϑ_{k+1} can be computed using the bounds for eigenvalues of the iteration matrix, given by Wathen [29], and then using the recursive definition of the Chebyshev polynomials.

```
Set $D = diag(M)$
Set relaxation parameter $\omega$
Compute $\mathbf{g} = \omega D^{-1}\hat{\mathbf{b}}$
Set $S = (I - \omega D^{-1}M)$ (this can be used implicitly)
Set $\mathbf{z}_0 = 0$ and $\mathbf{z}_1 = S\mathbf{z}_{k-1} + \mathbf{g}$
$c_0 = 2$ and $c_1 = \omega$
for $k = 2, \dots, l$ do
    $c_{k+1} = \omega c_k - \frac{1}{4}c_{k-1}$
    $\vartheta_{k+1} = \omega\frac{c_k}{c_{k+1}}$
    $\mathbf{z}_{k+1} = \vartheta_{k+1}(S\mathbf{z}_k + \mathbf{g} - \mathbf{z}_{k-1}) + \mathbf{z}_{k-1}$
end for
```

Algorithm 1. Chebyshev semi-iterative method for a number of l steps.

We now turn our attention to the approximation of the Schur complement, S_0. We want a linear operator that has the action of $(KM^{-1}K^T)^{-1} = K^{-T}MK^{-1}$. It is well known (see, for example, [8]) that a fixed number of multigrid V-cycles is an efficient preconditioner for the stiffness matrix, K. However, if $\hat{K}$ is an effective preconditioner for K, then $\hat{K}^2$ is not necessarily an effective preconditioner for K^2, which is essentially the situation we have here. Fortunately, for our case Braess and Peisker [22] showed that $KM^{-1}K^T$ is spectrally equivalent to $\hat{K}M^{-1}\hat{K}^T$, where $\hat{K}^{-1}$ denotes a fixed number of multigrid V-cycles. In the results that follow we use the Trilinos ML package [10] – an algebraic multigrid method (AMG) routine.

The second preconditioner we discuss here is a block triangular preconditioner which was proposed by Rees and Stoll in [24]. This preconditioner has the form

$$\mathcal{P}_{BT} = \begin{bmatrix} \gamma_0 A_0 & 0 & 0 \\ 0 & \beta\gamma_0 A_0 & 0 \\ -K & M & -S_0 \end{bmatrix},$$

where γ_0 is a scaling factor defined below. If we apply this preconditioner (on the left) to the matrix $\mathcal{K}$ the resulting matrix will be, in general, nonsymmetric, but as Bramble and Pasciak showed in [5], a non-standard inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ can be introduced such that the preconditioned matrix $\mathcal{P}^{-1}\mathcal{K}$ is symmetric and positive definite in this inner product. Based on this observation a Bramble–Pasciak version of the CG method can be implemented, see Algorithm 2.

```
Given $\mathbf{x}_0 = 0$, set $\mathbf{r}_0 = \mathcal{P}^{-1}(\mathbf{b} - \mathcal{K}\mathbf{x}_0)$ and $\mathbf{p}_0 = \mathbf{r}_0$
for $k = 0, 1, \dots$ do
    $\alpha = \frac{\langle \mathbf{r}_k, \mathbf{r}_k\rangle_{\mathcal{H}}}{\langle \mathcal{P}^{-1}\mathcal{K}\mathbf{p}_k, \mathbf{p}_k\rangle_{\mathcal{H}}}$
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha\mathbf{p}_k$
    $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha\mathcal{P}^{-1}\mathcal{K}\mathbf{p}_k$
    $\beta = \frac{\langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1}\rangle_{\mathcal{H}}}{\langle \mathbf{r}_k, \mathbf{r}_k\rangle_{\mathcal{H}}}$
    $\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta\mathbf{p}_k$
end for
```

Algorithm 2. Non-standard inner-product CG.

The non-standard inner product is given by $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{H}_{BT}} = \mathbf{x}^T \mathcal{H}_{BT} \mathbf{y}$, where

$$\mathcal{H}_{BT} = \begin{bmatrix} M - \gamma_0 A_0 & 0 & 0 \\ 0 & \beta(M - \gamma_0 A_0) & 0 \\ 0 & 0 & -S_0 \end{bmatrix}.$$

Note that this only defines an inner product if the diagonal blocks are symmetric and positive definite. This is typically a drawback of the Bramble–Pasciak CG as this might involve an expensive calculation to determine an appropriate scaling parameter, γ_0, which ensures positive-definiteness.

Recently, Rees and Stoll showed in [24] that for A_0 as defined above, the scaling can be done trivially and good performance can be observed. The condition that $M - \gamma_0 A_0$ is positive definite is equivalent to $\frac{1}{\gamma_0} A_0^{-1} M - I$ being positive definite, and hence we need to know the smallest eigenvalue of $A_0^{-1} M$. For the case where A_0 is a fixed number of steps of the Chebyshev semi-iteration the largest and smallest eigenvalues of $A_0^{-1} M$ are known analytically for any given number of iterations – these values are tabulated in [24]. An appropriate scaling γ_0 with the required properties can then simply be chosen.

We want to discuss the Bramble–Pasciak CG in more detail as the efficient implementation of the Bramble–Pasciak method can be quite subtle (see [7, 26]). For reasons of convenience we will go back to the 2×2 saddle point formulation where A_0 is the preconditioner for the $(1, 1)$ block and S_0 the Schur-complement preconditioner. The computation of α in Algorithm 2 (line 3) can be done in the following way

$$\langle r_k, r_k \rangle_{\mathcal{H}} = r_k^T \begin{bmatrix} A A_0^{-1} \tilde{r}_k^{(1)} - \tilde{r}_k^{(1)} \\ B A_0^{-1} \tilde{r}_k^{(1)} - \tilde{r}_k^{(2)} \end{bmatrix} = \langle r_k^{(1)}, (\mathcal{K} r_k)^{(1)} - \tilde{r}_k^{(1)} \rangle - \langle r_k^{(2)}, r_k^{(2)} \rangle \tag{32}$$

and

$$\begin{aligned} \langle \mathcal{P}^{-1} \mathcal{K} p_k, p_k \rangle_{\mathcal{H}} &= p_k^T \begin{bmatrix} A A_0^{-1} - I & 0 \\ B A_0^{-1} & -I \end{bmatrix} \begin{bmatrix} \hat{p}_k^{(1)} \\ \hat{p}_k^{(2)} \end{bmatrix} \\ &= \langle (\mathcal{K} p_k)^{(1)}, A_0^{-1} (\mathcal{K} p_k)^{(1)} \rangle - \langle p_k^{(1)}, (\mathcal{K} p_k)^{(1)} \rangle - \langle p_k^{(2)}, (\mathcal{K} p_k)^{(2)} \rangle \end{aligned} \tag{33}$$

where the indices in $r_k^{(1)}$ and $r_k^{(2)}$ and other vectors stand for the blocks corresponding to the components of the saddle point matrix, and the vector $\tilde{r}_k$ corresponds to the unpreconditioned residual. It can now easily be seen that we never explicitly need A_0 or S_0 as the matrices are usually not given, e.g. A_0^{-1} represents a multigrid operator. We can now easily compute line 6 in Algorithm 2. It has to be noted that both (32) and (33) require the computation of a product with $\mathcal{K}$ which should be avoided. Therefore, we will use the relation

$$\mathcal{K} p_{k+1} = \mathcal{K} r_{k+1} + \beta \mathcal{K} p_k$$

to only compute one matrix vector product per iteration. As a result the Bramble–Pasciak CG method only needs one more multiplication with the matrix B in comparison to MINRES with block-diagonal preconditioning.

4.2. With bound constraints

We now want to present a numerical scheme to solve problem (5). The method we want to analyze here is a primal-dual active set method introduced in [3]. For reasons of convenience, we use a new Lagrange multiplier $\boldsymbol{\mu}$ instead of $\boldsymbol{\mu}_a$ and $\boldsymbol{\mu}_b$ which is defined as follows

$$\boldsymbol{\mu} := \boldsymbol{\mu}_a - \boldsymbol{\mu}_b = \beta M\mathbf{u} + M\boldsymbol{\lambda}. \tag{34}$$

Then we get for the optimal control

$$(\mathbf{u}^*)_i \begin{cases} = (\underline{\mathbf{u}}_{\mathbf{a}})_i & \text{if } (\boldsymbol{\mu})_i > 0 \\ \in U_{ad} & \text{if } (\boldsymbol{\mu})_i = 0 \\ = (\overline{\mathbf{u}}_{\mathbf{b}})_i & \text{if } (\boldsymbol{\mu})_i < 0. \end{cases} \tag{35}$$

The quantity $\mathbf{u}^* - \boldsymbol{\mu}$ is an indicator whether a constraint is active or inactive and based on this an active set strategy can be implemented. For a general introduction to active set methods we refer to [11, 20] and in the particular case of a primal-dual active set strategy for PDE constrained optimization we refer to [3, 18, 28].

In more detail, we define the active sets as

$$\begin{aligned} \mathcal{A}_+ &= \{i \in \{1, 2, \ldots, N\} : (\mathbf{u}^* - \boldsymbol{\mu})_i > (\overline{\mathbf{u}}_{\mathbf{b}})_i\} \qquad (36) \\ \mathcal{A}_- &= \{i \in \{1, 2, \ldots, N\} : (\mathbf{u}^* - \boldsymbol{\mu})_i < (\underline{\mathbf{u}}_{\mathbf{a}})_i\} \qquad (37) \\ \mathcal{A}_I &= \{1, 2, \ldots, N\} \setminus (\mathcal{A}_+ \cup \mathcal{A}_-) \qquad (38) \end{aligned}$$

and note that the following conditions have to hold in each step of an iterative procedure

$$\begin{aligned} M\mathbf{y}^{(k)} - M\bar{\mathbf{y}} - K^T\boldsymbol{\lambda}^{(k)} &= 0 \qquad (39) \\ -K\mathbf{y}^{(k)} + M\mathbf{u}^{(k)} &= d \qquad (40) \\ \beta M\mathbf{u}^{(k)} + M\boldsymbol{\lambda}^{(k)} - \mu^{(k)} &= 0 \qquad (41) \\ \boldsymbol{\mu}^{(k)} &= 0 \text{ on } \mathcal{A}_I^{(k)} \qquad (42) \\ \mathbf{u}^{(k)} &= \underline{\mathbf{u}}_{\mathbf{a}} \text{ on } \mathcal{A}_-^{(k)} \qquad (43) \\ \mathbf{u}^{(k)} &= \overline{\mathbf{u}}_{\mathbf{b}} \text{ on } \mathcal{A}_+^{(k)}. \qquad (44) \end{aligned}$$

The full numerical scheme is summarized in Algorithm 3.

```
Choose initial values for $\mathbf{u}^{(0)}$, $\mathbf{y}^{(0)}$. $\boldsymbol{\lambda}^{(0)}$ and $\boldsymbol{\mu}^{(0)}$
for $k = 1, 2, \ldots$ do
    Set the active sets $\mathcal{A}_+^{(k)}$, $\mathcal{A}_-^{(k)}$ and $\mathcal{A}_I^{(k)}$ as given in (36), (37) and (38)
    if $k > 1$, $\mathcal{A}_+^{(k)} = \mathcal{A}_+^{(k-1)}$, $\mathcal{A}_-^{(k)} = \mathcal{A}_-^{(k-1)}$, and $\mathcal{A}_I^{(k)} = \mathcal{A}_I^{(k-1)}$ then
        STOP (Algorithm converged)
    else
        Solve (39) to (44)
    end if
end for
```

Algorithm 3. Active set algorithm.

Solving (39), (40) and (41) results in the linear system

$$\begin{bmatrix} M & 0 & -K \\ 0 & \beta M & M \\ -K & M & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}^{(k)} \\ \mathbf{u}^{(k)} \\ \boldsymbol{\lambda}^{(k)} \end{bmatrix} = \begin{bmatrix} M\bar{\mathbf{y}} \\ \boldsymbol{\mu}^{(k)} \\ 0 \end{bmatrix}. \tag{45}$$

Using a technique given in [9, 27] the linear system (45) can be reduced to the following linear system

$$\begin{bmatrix} M & 0 & -K \\ 0 & \beta M^{\mathcal{A}_I^{(k)},\mathcal{A}_I^{(k)}} & M^{\mathcal{A}_I^{(k)},:} \\ -K & M^{:,\mathcal{A}_I^{(k)}} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}^{(k)} \\ \mathbf{u}^{\mathcal{A}_I^{(k)}} \\ \boldsymbol{\lambda}^{(k)} \end{bmatrix} = \begin{bmatrix} M\bar{\mathbf{y}} \\ -\beta M^{\mathcal{A}_I^{(k)},\mathcal{A}_+^{(k)}}\overline{\mathbf{u}}_{\mathbf{b}} - \beta M^{\mathcal{A}_I^{(k)},\mathcal{A}_-^{(k)}}\underline{\mathbf{u}}_{\mathbf{a}} \\ -M^{:,\mathcal{A}_+^{(k)}}\overline{\mathbf{u}}_{\mathbf{b}} - M^{:,\mathcal{A}_-^{(k)}}\underline{\mathbf{u}}_{\mathbf{a}} \end{bmatrix} \tag{46}$$

using the fact that the control is known on both $\mathcal{A}_-^{(k)}$ and $\mathcal{A}_+^{(k)}$. Once, the system (46) is solved, we can update the Lagrange multipliers associated with the sets $\mathcal{A}_+^{(k)}$ and $\mathcal{A}_-^{(k)}$

$$\begin{aligned} \boldsymbol{\mu}^{\mathcal{A}_+^{(k)}} &= \beta M^{\mathcal{A}_+^{(k)},\mathcal{A}_I^{(k)}}\mathbf{u}^{\mathcal{A}_I^{(k)}} + \beta M^{\mathcal{A}_+^{(k)},\mathcal{A}_+^{(k)}}\overline{\mathbf{u}}_{\mathbf{b}} + \beta M^{\mathcal{A}_+^{(k)},\mathcal{A}_-^{(k)}}\underline{\mathbf{u}}_{\mathbf{a}} + M^{\mathcal{A}_+^{(k)},:}\boldsymbol{\lambda}^{(k)} \\ \boldsymbol{\mu}^{\mathcal{A}_-^{(k)}} &= \beta M^{\mathcal{A}_-^{(k)},\mathcal{A}_I^{(k)}}\mathbf{u}^{\mathcal{A}_I^{(k)}} + \beta M^{\mathcal{A}_-^{(k)},\mathcal{A}_+^{(k)}}\overline{\mathbf{u}}_{\mathbf{b}} + \beta M^{\mathcal{A}_-^{(k)},\mathcal{A}_-^{(k)}}\underline{\mathbf{u}}_{\mathbf{a}} + M^{\mathcal{A}_-^{(k)},:}\boldsymbol{\lambda}^{(k)}. \end{aligned} \tag{47}$$

The linear system (46) can now be solved using either the block-diagonal or block-triangular preconditioners presented in Section 4.1 as this represents an unconstrained problem on the free variables represented in $\mathcal{A}_I$. In [9] multigrid approaches are presented in order to solve (46). It has to be noted that in comparison to the unconstrained problem the cost per iteration for one iteration of the active set method corresponds to solving the unconstrained problem. But the active set method benefits in subsequent steps from having a good initial guess which reduces the number of iterations needed to solve the linear system in later iterations.

A convergence criterion for the method is given in [3], i. e., if the active sets stay unchanged in two consecutive steps the method has found a local minimum and the algorithm can be terminated. In [16, 27] it is demonstrated that the active set method presented here is a semi-smooth Newton method that under certain conditions gives superlinear convergence. Stoll and Wathen [27] showed that this method can also be derived when we start from a projected gradient approach with Newton acceleration as given in [4]. A projected gradient method with steepest descent direction can also be used but the cost is comparable to the active set method for each iteration step but the convergence is much slower (see [27]).

5. NUMERICAL RESULTS

We illustrate our method using the following example. Let $\Omega = [0,1]^m$, where $m = 2, 3$, and consider the problem

$$\min_{y,u} \frac{1}{2}\|y - \bar{y}\|^2_{L_2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L_2(\Omega)}$$

$$\text{s.t.} \qquad -\nabla^2 y = u \qquad \text{in } \Omega \tag{48}$$

$$y = \bar{y} \qquad \text{on } \Gamma \tag{49}$$

where

$$\bar{y} = \begin{cases} -x_1 \exp\left(-\left((x_1-\frac{1}{2})^2+(x_2-\frac{1}{2})^2\right)\right) & \text{if } (x_1,x_2)\in[0,1]^2 \\ -x_1 \exp\left(-\left((x_1-\frac{1}{2})^2+(x_2-\frac{1}{2})^2+(x_3-\frac{1}{2})^2\right)\right) & \text{if } (x_1,x_2,x_3)\in[0,1]^3. \end{cases}$$

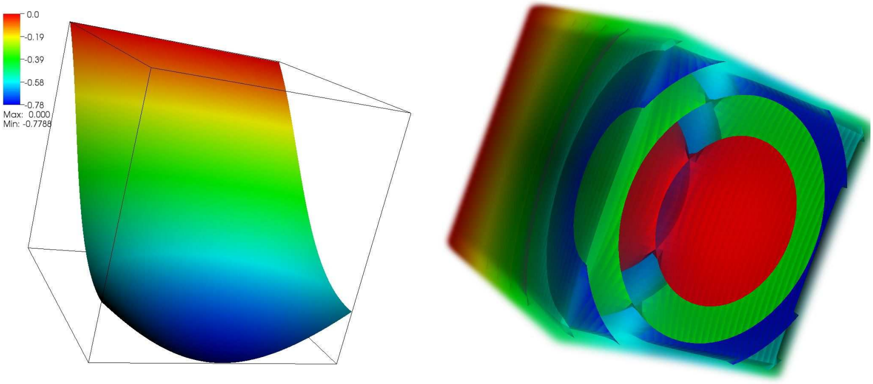
The bounds $\underline{\mathbf{u}}_{\mathbf{a}}$ and $\overline{\mathbf{u}}_{\mathbf{b}}$ are defined as follows

$$\underline{\mathbf{u}}_{\mathbf{a}} = \begin{cases} -0.35 & \text{if } x_1 < 0.5 \\ -0.4 & \text{otherwise} \end{cases}$$

and

$$\overline{\mathbf{u}}_{\mathbf{b}} = \begin{cases} -0.1\exp\left(-\left(x_1^2+x_2^2\right)\right) & \text{if } (x_1,x_2)\in[0,1]^2 \\ -0.1\exp\left(-\left(x_1^2+x_2^2+x_3^2\right)\right) & \text{if } (x_1,x_2,x_3)\in[0,1]^3. \end{cases}$$

Figure 1 illustrates the the desired state. We discretize the optimality system using $Q1$ finite elements using dealii [1]. The tolerance for both methods is given by 10^{-6} where we check for both methods the unpreconditioned relative residual.



(a) Desired state in 2D

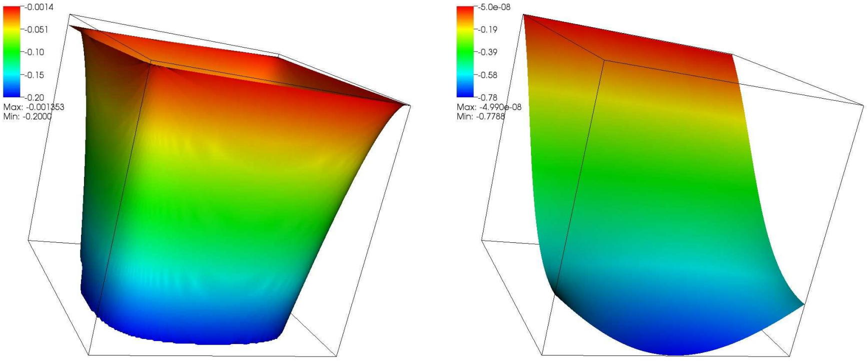
(b) Desired state in 3D

Fig. 1. Desired state $\bar{y}$.

Figure 2 shows the state and the control in two dimensions for the optimal control problem with control constraints for $\beta = 10^{-2}$.

For the active set method, Bergounioux et al. [3] use different start-up conditions. We employ only that which proved best for the examples analyzed in [3], namely

$$\begin{cases} \mathbf{u}^{(0)} = \overline{\mathbf{u}}_{\mathbf{b}} \\ K\mathbf{y}^{(0)} = M\mathbf{u}^{(0)} \\ K^T\boldsymbol{\lambda}^{(0)} = M\mathbf{y}^{(0)} - M\bar{\mathbf{y}} \\ \boldsymbol{\mu}^{(0)} = \beta M\mathbf{u}^{(0)} + M\boldsymbol{\lambda}^{(0)}. \end{cases} \tag{50}$$



(c) Control

(d) State

Fig. 2. Control and state for $\beta = 10^{-2}$.

We now want to introduce the overall setup that is used for our computations. For A_0 we will use 10 steps of the Chebyshev semi-iterative method. The Schur complement is approximated by $S_0 = \hat{K}M^{-1}\hat{K}^T$ where $\hat{K}$ represents two V cycles of the ML AMG [10] with 10 steps of a Chebyshev smoother. As a basis for our computations we use dealii [1] a C++ framework for finite element calculations.

Table 1. Iteration numbers and CPU times for different mesh sizes in two dimensions.

N	BPCG	MINRES	AS(BPCG)	t(BPCG)	t(MINRES)	t(AS)
289	8	12	3(21)	0.02	0.03	0.07
1089	8	10	3(23)	0.08	0.09	0.26
4225	8	12	4(37)	0.32	0.44	1.56
16641	9	13	4(44)	1.62	2.17	8.23
66049	10	16	4(52)	8.7	13.03	46.46
263169	12	21	4(61)	43.54	71.93	227.18
1050625	15	38	4(88)	222.9	528.04	1305.17

Table 1 shows the results in two dimensions obtained for different mesh sizes. The number of Bramble–Pasciak CG and MINRES iterations and CPU times are shown, as well as the number of outer active set iterations together with the total number of Bramble–Pasciak CG solves needed for the linear systems within the active set method, and the CPU times in this case. Table 2 shows the equivalent data in three dimensions.

Table 2. Iteration numbers and CPU times for different mesh sizes in three dimensions.

N	BPCG	MINRES	AS(BPCG)	t(BPCG)	t(MINRES)	t(AS)
125	7	9	2(13)	0.01	0.01	0.02
729	7	10	2(14)	0.08	0.11	0.18
4913	7	10	2(15)	0.64	0.83	1.5
35937	7	10	3(21)	6.15	8	19.7
274625	7	10	3(22)	52.19	68.58	173.94
2146689	7	12	4(32)	445.05	693.98	2128.25

Figure 3 shows the iterations numbers of MINRES, the Bramble–Pasciak CG, and the total number of Bramble–Pasciak CG iterations for a number of smaller matrices ($N = 729, 4913, 35937$) and different β values. All results are for the previously described setup in three dimensions. It can be seen that the iteration numbers are constant with respect to mesh-size but go up once the regularization parameter β is decreasing.

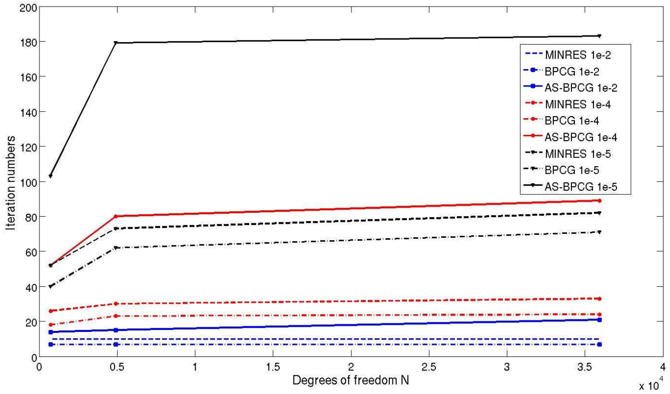


Fig. 3. Iterations for MINRES, Bramble–Pasciak CG, and the total number of Bramble–Pasciak iterations for the active set method.

6. CONCLUSION

In this paper we have illustrated how all-at-once methods can be employed to solve problems from PDE-constrained optimization. In particular, we showed that both problems with and without control constraints lead to linear systems in saddle point form and we presented efficient preconditioning strategies for both problems. We have discussed implementation issues that arise from using available finite element packages as well as looking at different boundary conditions as part of the state equation. We illustrated the efficiency and competitiveness of our approach.

ACKNOWLEDGEMENT

This publication is partially based on work supported by Award No. KUK-C1-013-04, made by King Abdullah University of Science and Technology (KAUST). This paper summarises the talk given by the third author at Algorithmy 2009 in Podbanské, Slovakia.

(Received March 3, 2010)

REFERENCES

[1] W. Bangerth, R. Hartmann, and G. Kanschat: deal.II—a general-purpose object-oriented finite element library. ACM Trans. Math. Software *33* (2007), 24, 27.

[2] M. Benzi, G. H. Golub, and J. Liesen: Numerical solution of saddle point problems. Acta Numer. *14* (2005), 1–137.

[3] M. Bergounioux, K. Ito, and K. Kunisch: Primal-dual strategy for constrained optimal control problems. SIAM J. Control Optim. *37* (1999), 1176–1194 (electronic).

[4] D. P. Bertsekas: Projected Newton methods for optimization problems with simple constraints. SIAM J. Control Optim. *20* (1982), 221–246.

[5] J. H. Bramble and J. E. Pasciak: A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. Math. Comp *50* (1988), 1–17.

[6] S. S. Collis and M. Heinkenschloss: Analysis of the Streamline Upwind/Petrov Galerkin Method Applied to the Solution of Optimal Control Problems. Tech. Rep. TR02–01, Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005–1892, 2002.

[7] H. C. Elman: Multigrid and Krylov subspace methods for the discrete Stokes equations. In: Seventh Copper Mountain Conference on Multigrid Methods (N. D. Melson, T. A. Manteuffel, S. F. McCormick, and C. C. Douglas, eds.), Vol. CP 3339, Hampton 1996, NASA, pp. 283–299.

[8] H. C. Elman, D. J. Silvester, and A. J. Wathen: Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics. Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2005.

[9] M. Engel and M. Griebel: A multigrid method for constrained optimal control problems: SFB-Preprint 406, Sonderforschungsbereich 611, Rheinische Friedrich-Wilhelms-Universität Bonn, 2008. Submitted.

[10] M. Gee, C. Siefert, J. Hu, R. Tuminaro, and M. Sala: ML 5.0 smoothed aggregation user's guide. Tech. Rep. SAND2006-2649, Sandia National Laboratories, 2006.

[11] P. E. Gill, W. Murray, and M. H. Wright: Practical Optimization. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], London, 1981.

[12] G. H. Golub and R. S. Varga: Chebyshev semi-iterative methods. successive over-relaxation iterative methods, and second order Richardson iterative methods. I. Numer. Math. *3* (1961), 147–156.

[13] G. H. Golub and R. S. Varga: Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods. II. Numer. Math. *3* (1961), 157–168.

[14] A. Greenbaum: Iterative Methods for Solving Linear Systems. Vol. 17 of Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia 1997.

[15] M. R. Hestenes and E. Stiefel: Methods of conjugate gradients for solving linear systems. J. Res. Nat. Bur. Stand *49* (1952), 409–436 (1953)???.

[16] M. Hintermüller, K. Ito, and K. Kunisch: The primal-dual active set strategy as a semismooth Newton method. SIAM J. Optim. *13* (2002), 865–888.

[17] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich: Optimization with PDE Constraints. Mathematical Modelling: Theory and Applications. Springer-Verlag, New York 2009.

[18] K. Ito and K. Kunisch: Lagrange multiplier approach to variational problems and applications. Vol. 15 of Advances in Design and Control, Society for Industrial and Applied Mathematics (SIAM), Philadelphia 2008.

[19] M. F. Murphy, G. H. Golub, and A. J. Wathen: A note on preconditioning for indefinite linear systems. SIAM J. Sci. Comput. *21* (2000), 1969–1972.

[20] J. Nocedal and S. J. Wright: Numerical Optimization. (Springer Series in Operations Research.) Springer-Verlag, New York 1999.

[21] C. C. Paige and M. A. Saunders: Solutions of sparse indefinite systems of linear equations. SIAM J. Numer. Anal. *12* (1975), 617–629.

[22] P. Peisker and D. Braess: A conjugate gradient method and a multigrid algorithm for Morley's finite element approximation of the biharmonic equation. Numer. Math. *50* (1987), 567–586.

[23] T. Rees, H. S. Dollar, and A. J. Wathen: Optimal solvers for PDE-constrained optimization. SIAM J. Sci. Comput. *32* (2010), 271–298.

[24] T. Rees and M. Stoll: Block triangular preconditioners for PDE constrained optimization. Numer. Linear Algebra Appl., (2009), to appear.

[25] Y. Saad: Iterative Methods for Sparse Linear Systems. Society for Industrial and Applied Mathematics, Philadelphia 2003.

[26] M. Stoll: Solving Linear Systems Using the Adjoint. PhD. Thesis, University of Oxford 2009.

[27] M. Stoll and A. Wathen: Preconditioning for active set and projected gradient methods as semi-smooth Newton methods for PDE-constrained optimization with control constraints. Submitted.

[28] F. Tröltzsch: Optimale Steuerung partieller Differentialgleichungen: Theorie, Verfahren und Anwendungen. Vieweg Verlag, Wiesbaden 2005.

[29] A. J. Wathen: Realistic eigenvalue bounds for the Galerkin mass matrix. IMA J. Numer. Anal. *7* (1987), 449–457.

[30] A. J. Wathen and T. Rees: Chebyshev semi-iteration in preconditioning for problems including the mass matrix. Electron. Trans. Numer. Anal. *34* (2008–2009), 125–135.

Tyrone Rees, Mathematical Institute, University of Oxford, 24–29 St. Giles', Oxford, OX1 3LB. United Kingdom.
e-mail: tyrone.rees@maths.ox.ac.uk

Martin Stoll, Oxford Centre for Collaborative Applied Mathematics, Mathematical Institute, 24–29 St Giles', Oxford, OX1 3LB. United Kingdom
e-mail: martin.stoll@maths.ox.ac.uk

Andy Wathen, Mathematical Institute, University of Oxford, 24–29 St. Giles', Oxford, OX1 3LB. United Kingdom.
e-mail: andy.wathen@maths.ox.ac.uk